Interactive Digital Twins Framework for Asset Management Through Internet

Citation for published version:
Kong, CW, Harper, S, Mitchell, D, Blanche, J, Lim, T & Flynn, D 2021, Interactive Digital Twins Framework for Asset Management Through Internet. in *2020 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT).*, 9345890, IEEE, 2020 IEEE Global Conference on Artificial Intelligence and Internet of Things, 12/12/20. https://doi.org/10.1109/GCAIoT51063.2020.9345890

Digital Object Identifier (DOI):
10.1109/GCAIoT51063.2020.9345890

Link:
Link to publication record in Heriot-Watt Research Portal

Document Version:
Peer reviewed version

Published In:
2020 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)

Interactive Digital Twins Framework for Asset Management Through Internet

Leo Chi Wai Kong
Smart Systems Group
School of Engineering & Physical Sciences
Heriot-Watt University
Edinburgh, UK
cwk2@hw.ac.uk

Sam Harper
Smart Systems Group
School of Engineering & Physical Sciences
Heriot-Watt University
Edinburgh, UK
sh324@hw.ac.uk

Daniel Mitchell
Smart Systems Group
School of Engineering & Physical Sciences
Heriot-Watt University
Edinburgh, UK
dm68@hw.ac.uk

Jamie Blanche
Smart Systems Group
School of Engineering & Physical Sciences
Heriot-Watt University
Edinburgh, UK
J.Blanche@hw.ac.uk

Theodore Lim
Smart Systems Group
School of Engineering & Physical Sciences
Heriot-Watt University
Edinburgh, UK
T.Lim@hw.ac.uk

David Flynn
Smart Systems Group
School of Engineering & Physical Sciences
Heriot-Watt University
Edinburgh, UK
D.Flynn@hw.ac.uk

Abstract— **Digitalization is influencing the design, operation and management, as well as planning functions for products and services across a myriad of industries. In our research we focus on the specific needs and challenges in the asset management of remote critical infrastructure. We propose a single Digital Twin framework which can synchronize the data and communication protocols across multiple devices to support exchanging data between the physical world and the cyber world under any scenario, anywhere and at any time. Our framework can support the synchronization of 1000 different sensors and actuators. The results of our Digital Twin are demonstrated using embedded, front-end sensing for offshore energy assets. It can filter and translate complex data and messages from any embedded sensor and operating system. Furthermore, we show how a complete Digital Twin framework allows end-users to simulate future events capturing the interactions between the environment, people and assets, enabling a better understanding of operational risks and remaining useful life of assets.**

Keywords— Cyber Physical System, Digital Twin, Embedded Sensing, Asset Management, Internet of Things

I. Introduction

A Digital Twin (DT) can be described as the mirror or the copy of a physical object, a digital representation of the physical systems and devices with their environment and lifecycle at any time, and builds an interaction between the physical and virtual worlds [1]–[8]. A DT is commonly used to reflect the performance, health and maintenance data from the connected physical assets [9]. DT technologies continuously collect data from IoT sensing devices to simulate dynamic physical objects and environment, and that data is used to enhance the virtual models for reflecting the run-time situation of the physical world [2], [10]–[13]. Ideally, having high-fidelity virtual models that reflect the physical entity in detail is preferred [11], [12]. Overall, the virtual representations within the DT must be able to accurately reflect the feedback from the physical world in real-time [5], [14]. Subsequently, the digital models used are required to be realistic because they also represent the real physical relationship between the hardware and the environment, such as conditions and constraints [3], [12]. Bi-directional data transmission and the interaction between virtual models and the physical assets are key aspects of any DT. The DT of an asset must be able to display the current status, movement and behavior of the asset with a mono-directional data flow set up. However, DTs can be used to reversely control the connected physical asset with a complete bi-directional data flow set up [4].

The concept of monitoring and controlling physical assets through DTs are demonstrated with examples, and this is the optimal way to improve operational efficiency and quality [5], [12], [15]. For this, a DT framework requires a method to synchronize and exchange multimodal data at run-time between physical and digital assets. Operators can plan and review potential concept behaviors on the virtual model using both real and generative data prior to executing a teleoperation [3], [5], [12], [16], [17]. Visualization and an effective user interface would aid users to interact with the system and the asset, and to complete the features of run-time monitoring and record-and-replay [12], [18], [19]. A generic and extendable DT framework should be able to accept multiple sources of data, interoperate across multiple assets and be applied at different scales and scenario implementations [2], [3], [6], [12], [20]. The primary purpose of a DT is to provide data-driven decision support to end-users [5], [19], [21]. Therefore, the DT framework is suggested to have communication channels for exchanging data and commands and an interface to reflect the physical side data as well as demonstrating concept behaviors of the asset. More

This work was supported by funding from EPSRC under grants EP/R026173/1 and EP/P009743/1.

importantly, the framework should be designed to accept multiple physical elements for mirroring the physical world virtually.

Data driven condition monitoring has been proven to be a strong, reliable and cost saving method to allow end users to make operational decisions based on the run-time status of their assets, where the information acquired is used to monitor the health status of the target asset [22]–[24]. Machine learning is a method to translate and fuse the data collected to human-readable, meaningful information [22]. However, the current state of condition monitoring systems cannot display large and complex data sets efficiently, alongside a lack of communication between sub-systems [25].

The use of DT technologies for management of assets can result in significant savings in costs, as the DT is able to monitor an assets status, leading to improvements in work environment safety, efficiency and product quality [1], [9], [15], [17], [18]. DTs represent a direct solution to solve the insufficiencies in quality analysis and decision-making as complex systems become more realistic and easier to monitor and control. Consequently, DTs offer operators a better understanding of their physical hardware through the application of digital technologies within the existing DT examples [1], [2], [6], [16].

This paper presents how distributed embedded sensor networks can be used to support a Digital Twin framework, which can synchronize the data and communication protocols across multiple devices. This framework supports the exchanging of data between the physical world and the cyber world under any scenario, anywhere and at any time. The remainder of the paper is structured as follows; section II provides an overview of the experiment and use cases for evaluating the digital twin framework. Results and analysis are contained in section III where section IIIA contains the results for case I, an embedded system, section IIIB utilizes a FMCW sensor system and section IIIC demonstrates the results of a resilient robotics platform. Section IV presents the discussion and section V presents conclusions.

II. IMPLEMENTATION AND EVALUATION OF THE DIGITAL TWIN FRAMEWORK

Based on the current state of DT research, the DT framework presented in this paper has been designed to keep the essential elements of the existing DT model and to solve the current limitations. It provides two communication channels of data and teleoperation commands exchange between the physical and virtual environment; ensuring both sides are synchronized. The proposed framework involves the use of IoTs, network communication, communication protocol, data management and CAD models. This framework uses a non-intrusive, add-on method to tune physical assets as IoT enabled devices for communication. Low cost edge computing devices, such as Raspberry Pi and Nvidia Jetson could be used as a node of the network communication. A server located between the physical and the digital sides acts as a data exchange center to collate, encode and transmit data at low latency. Fig. 1 illustrates the architecture of the DT framework.

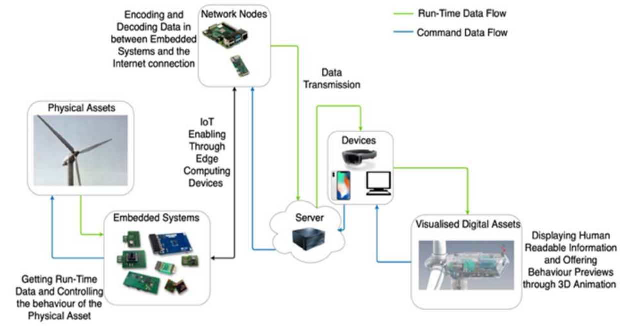

Fig. 1 The Concept of DT Framework Architecture

The client-side application has been designed to complete the DT for displaying information and being used as a user interface. This was planned to be a cross-platform application by using Unity 3D and C# to enable the application to be run on different devices and operating systems (Windows, Linux, macOS, Android, etc.). High fidelity CAD models were used to create the twin of physical assets in a digital format. All pertinent components of the real asset are included in the CAD model to generate digital animation reflecting the real-world situation.

This framework allows operators to plan and preview the behavior of their physical assets through the Graphical User Interface (GUI) as well as receiving run-time data from the physical world. Teleoperation commands will be sent to the physical workspace once the operator ascertains that the simulated behavior is acceptable. The edge computing device decodes commands and works as an actuator for redirecting them to the connected embedded system and for controlling the asset. To verify the capability of the DT framework to reflect the target workspace at run-time, we evaluated three use cases to compare using a visualized DT for the processing of embedded sensor system data. Tailor-made GUIs were provided to each use case, and both GUIs were connected to a common command channel for teleoperation.

A. *Use Case 1: Embedded System*

In use case 1, two Arduino boards had different setup and update rates for emulating a cooling system. One was programmed to obtain the temperature data from the workspace. The other was connected to a computer cooling fan utilizing a hall effect sensor for monitoring the run-time speed of the fan. End users needed to use Command Line Interface (CLI) or the built-in “Serial Monitor” from Arduino Integrated Development Environment (IDE) to receive the run-time data through the serial port. Each Arduino board was connected to a Raspberry Pi for IoT enabling, as shown in Fig. 2. No direct communication channel was given between those devices, both separate devices were enabled to connect and communicate to the DT server through the internet, and the Raspberry Pi converted the data and commands from DT framework protocols to the embedded system protocol as well.

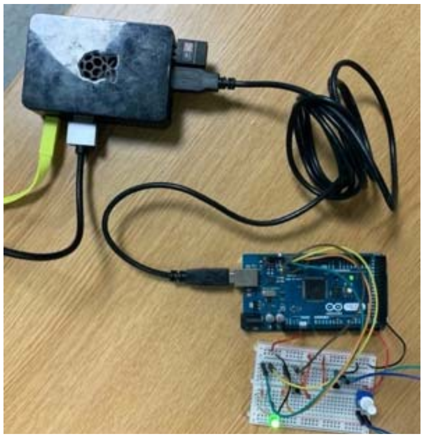

Fig. 2 Temperature Sensor Connected to Raspberry Pi 2B with a WIFI adaptor for IoT Enabling

B. Use Case 2: FMCW Sensor System

In use case 2, an integrated Frequency Modulated Continuous Wave (FMCW) radar was used. The sensor provides information such as range-to-target and surface properties, accompanied by previously inaccessible information of subsurface properties of porous and dielectric structures [26]–[28]. The utilized K-band FMCW system represents a multi-purpose sensor which can provide integrated information after a single target sweep. The project engineers decided to communicate with the radar device and analyze the returned data through MATLAB. A serial port was the communication tunnel of the radar, which was connected to a laptop PC acting as a network node. This allowed the radar and the server to exchange data and commands.

C. Use Case 3: Resilient Robotics Platform

In use case 3, with combining the same DT framework to the Husky A200 robot which was driven by the Robotic Operating System (ROS), the user interface would be able to display the run-time status of Husky A200, as well as self-prognostic messages. The user interface also allowed the operator to plan and preview the actions of the attached UR5 robotic arms before sending the commands to the real-world robot.

III. Result and Analysis

The DT framework was given a connectivity test, Table 1 shows that this interoperable DT framework can connect and synchronize more than 1000 sensors and actuators, depending on the platform and computing resource. It also proves that this framework is stable, extendable and generic for different platforms, systems and hardware architectures.

TABLE I. Comparison of the DT Performance on Different Platforms

Platform	CPU	Memory	Accepted sensors and actuators
Desktop PC	2 X Intel XEON (octa-core) @ 2.1GHz	64 GB	> 1800
MacBook Pro 2017	Intel Core i7 (qual-core) @ 2.9GHz	16 GB	850
Jetson TX 2	ARM Cortex-A57 (quad-core) @ 2GHz + NVIDIA Denver2 (dual-core) @ 2GHz	8 GB	500

A. Use Case 1: Embedded System

This scenario exemplified a small-scale cooling system under the add-on IoT enabling setting. This mock-up was to replicate the cooling system in a control room with a generator, which supports serial port communication. Both the cooling device and the temperature sensor worked independently. The programmed embedded systems used the standard “set-up” and “loop” Arduino IDE programming format, which allowed them to configure the General-Purpose Input/ Output pins. This was especially important for tweeting them to accept incoming signals or transmit signals through the output wires before relayed into the main workload. Inside the main workload loop, the systems continuously checked the parameters from sensors, buttons and self-status for variations in their work behaviors.

Digital simulation was used to display the run-time status of connected assets. The GUI from Fig. 3 shows the 3D animation and filtered information, they also demonstrate the concept of simplifying communication with multiple devices using a single interface. This GUI combined the data from the connected temperature sensor and cooling fan. Animation and text information displayed the run-time behavior and status from the DT GUI. The GUI displays the run-time rotation speed of the cooling fan with an animation on the virtual twin, in addition to showing the numerical data. The color scheme of the virtual asset helps identify with the run-time health status of the physical asset. The GUI acted as a communication bridge by linking two completely independent embedded systems through the internet and illustrated run-time information. Operators can preview the speed of the cooling before sending the controlling command to the physical workspace. The preview function was activated when the speed control slider was changed. Another virtual asset showed up for representing the upcoming function based on the speed slider. The teleoperation command could be sent to the real asset after the operator agreed and click the “send” button.

B. Use Case 2: FMCW Sensor System

The objective of this use case was to display filtered data graphically from the FMCW radar, with no user knowledge required, and where human readable and simple information was the key output. The result of each scan was displayed as a Fast Fourier Transform (FFT) chart, as illustrated in Fig. 4. This use case required end users to have a basic level of background knowledge about the relevant radar theory and the contrasts to be expected from the differing target materials. Fig. 4. displays the difference in return signal amplitude in the frequency domain for two targets of significantly different material composition: a metal plate and the planar surface of a balsa and glass fiber composite wind turbine blade. There is a clear contrast in return signal amplitude from BIN 5 onwards. BIN 5

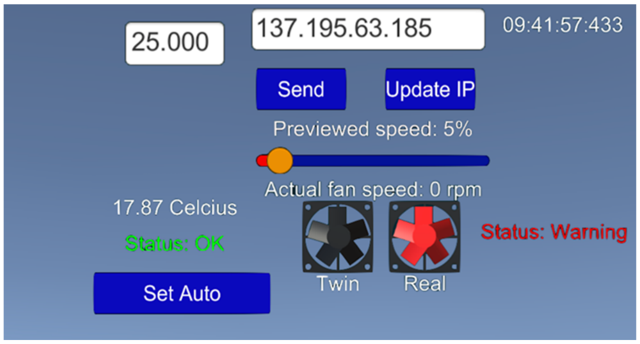

Fig. 3A screenshot of the DT GUI used animation and text messages to display the run-time parameters and workspace information, and it also changed the colour of the virtual asset for noticing users that faults have been detected. A twin of the virtual model was appeared for the preview of the control planning

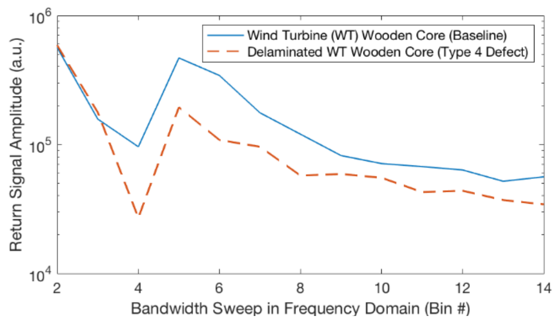

Fig. 4 Fourier transformed FMCW radar return signal displaying the contrast in asset integrity between an undamaged and damaged internal section of wind turbine blade structure

signifies the peak amplitude reflection for the target interface at 10 cm from the antenna, where data beyond BIN 5 represents the subsurface integrity of the wind turbine blade. The data extrapolated from the FMCW output successfully displayed the strength of returned signal amplitude by the distance of the object. However, operators of this system must be able to understand how to analyze the target specific data from the generated graphical output to attain real-world information of the target asset.

With the same DT framework, end-users could understand the status without the FFT chart on the bespoke DT GUI. A set of color schemes had been designed to reflect the structure of the scanned object. This interface also allowed users to set the number of FMCW scans and to start the operation remotely. Fig. 5 shows the DT interface that displayed a full-scale (1:1) wind turbine model, where the blade color was assigned following inspection with the FMCW radar. From this interface, a text box and a button for users to insert the number of times for scanning the object was provided. These commands were transmitted to the radar system and to initiate the scanning process. In this case, after 10 scans, the system analyzed and returned the result of the wind turbine blade sample, prompting the GUI to change the color of the blade on the DT.

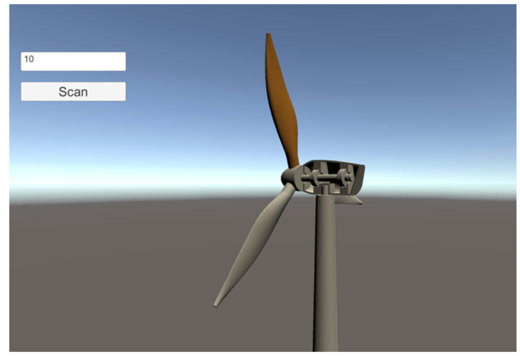

Fig. 5 A screenshot of the FMCW DT GUI changed color to illustrate the detected material of the wind turbine blade

The system also displayed text messages informing end users that the process had finished and what result had been acquired. Form the end user perspective, the system provides a single interface to interact with the radar sensor and review the analyzed result. This represents a more user-friendly GUI, allowing for a wider range of user proficiencies; displaying a color-coded output that does not require the operator to read complicated 2D FFT charts or depend on technical specialist knowledge.

C. Use Case 3: Resilient Robotics System

Human-Robot interaction was also investigated for this research. Robotic-driven manufacturing is an element of Industry 4.0 and the interaction between robot and human will be an increasingly important component for the next generation of manufacturing. For this scenario, a Clearpath Husky A200 wheeled robotic platform was used with two UR5 robotic arms attached. This robot has been designed and programmed for the use case of automated offshore infrastructure inspection with Offshore Robotics for Certification of Assets (ORCA) Hub. Researchers at the ORCA Hub had created a self-certification algorithm to monitor the run-time health status of the robot, which enabled the robot to self-certify its systems and determine if the robot able to finish its assigned mission. System ontology messages were communicated to the main system through the Robotic Operating System (ROS), which required operators to have a deep knowledge of the Linux operating system and CLI. Without a GUI and the method of data synchronization, operators were only able to monitor the mission status and behavior of all components, such as joint angles, wheel angles, voltage and current, through ROS topics.

A DT interface was developed as a visualized monitor and control panel for the Husky A200 platform equipped with dual manipulator arms. This DT was a single application to show the filtered information from the ontology messages and to animate the position of both UR5 arms of the real Husky robot. The color of the top chassis, and both arms of the virtual Husky, would be changed when the physical Husky detected internal system and hardware faults. The animation of the virtual Husky represented the run-time information of both robotic arms.

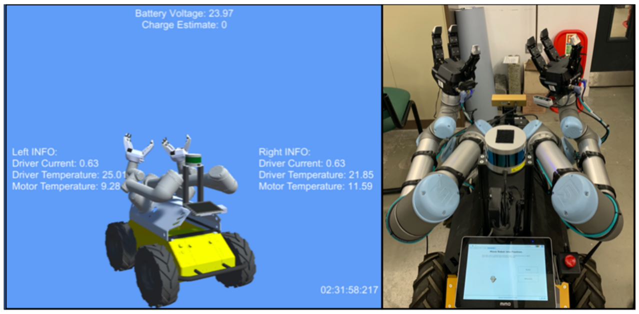

Fig. 6 A screenshot of the Husky A200 DT GUI and a picture of the physical robot, they demonstrate that both were synchronized through the DT framework

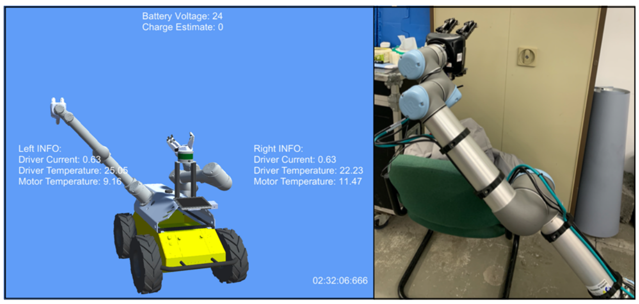

Fig. 7 A screenshot of the Husky A200 DT GUI and a picture of the physical UR5 robotic arm, they show that the virtual and the physical models were able to be synchronized using the DT framework in the real-time where the position and the joint angles of the robotic arm changed

The DT GUI displayed text messages showing latent system information, such as hardware voltage, current and temperature. In this use case, an external Raspberry Pi 3 was added and tasked with subscribing ROS topics from the Husky A200 main computer, which was running the ROS core. The Raspberry Pi 3 worked as an edge-computing device to process the raw data from ROS topics. It also connected and transmitted data to the DT server. The use of this additional device ensured all the computing resources would not be drained when extra tasks were added to Husky and demonstrates that missions that otherwise would have been interrupted by errors could be seen to be operating within normal parameters.

The DT preview function enabled users to remotely plan and control the UR5 arms of the Husky platform. The GUI displayed a semi-transparent twin on top of the virtual model of the robotic asset, which allowed users to understand the comparison between the current position and the planning position of the arms. This DT GUI integrated the information from ROS and simplified the controlling process with providing safety preview in a virtual environment. Commands would be sent to the aforementioned Raspberry Pi 3 for translation to a ROS topic.

Figs. 6 - 9 show that the virtual Husky was able to synchronize to the physical Husky A200. The DT GUI showed the run-time status of Husky by displaying text messages such

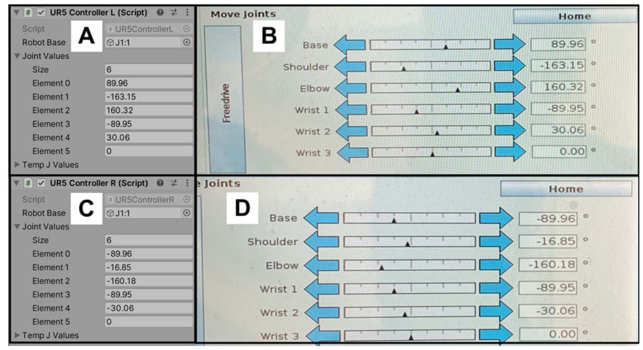

Fig. 8 The evidence of virtual and physical run-time parameter synchronized as the mission starts: A) the run-time parameters of the virtual model of the left UR5 arm, B) the control panel of the left UR5 arm with its run-time parameters, C) the run-time parameters of the virtual model of the right UR5 arm, D) the control panel of the right UR5 arm with its run-time parameters

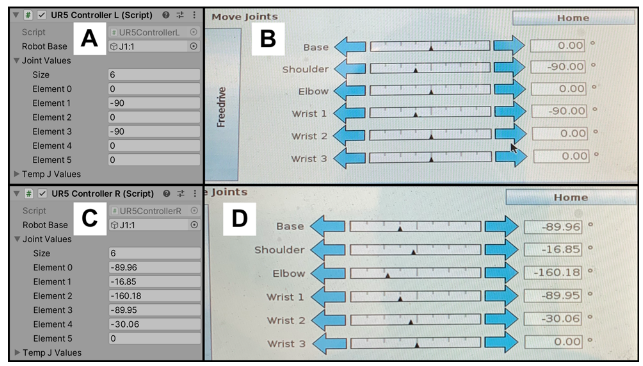

Fig. 9 The evidence of virtual and physical run-time parameter synchronized during the mission: A) the run-time parameters of the virtual model of the left UR5 arm, B) the control panel of the left UR5 arm with its run-time parameters, C) the run-time parameters of the virtual model of the right UR5 arm, D) the control panel of the right UR5 arm with its run-time parameters

as the information of the battery, temperature and current of two motors positioned inside the chassis. Subsequently, the ontology messages of Husky and the GUI also synchronized the run-time parameters of both UR5 arms Figs. 8 and 9 show the joint angles of the virtual Husky match the real parameters of both arms, where A and C from these figures show the parameters of each joint angle of the virtual model of the robotic arms. These parameters can be seen to match the run-time parameters of the real-world UR5 arms, which shown in B and D of Figs. 8 and 9. This demonstrates that the virtual model mirrored the actual robot actions and the GUI could repeatedly update those parameters from the real husky and use them to change the position of the arms in the virtual environment, where the arm position was changed from Fig. 6 to 7. Thus, the twin and the physical object were successfully synchronized, and the twin could reflect all run-time behaviors.

Fig. 10 illustrates that the GUI changed the color of the virtual Husky when an error was detected. This sensitive fault warning was displayed via text messages, which could allow users to acknowledge the current situation for run-time decision-

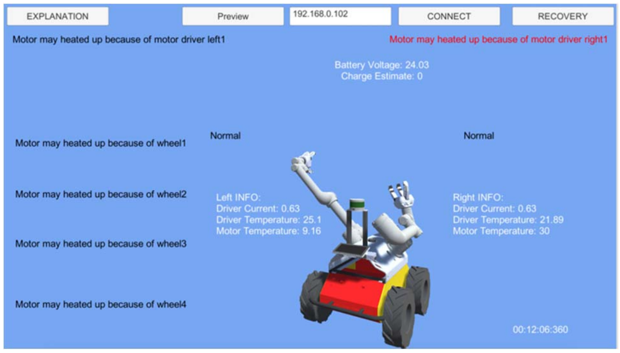

Fig. 10 A screenshot of the DT GUI shows that it changed the color of the virtual Husky and displayed error messages when faults were detected

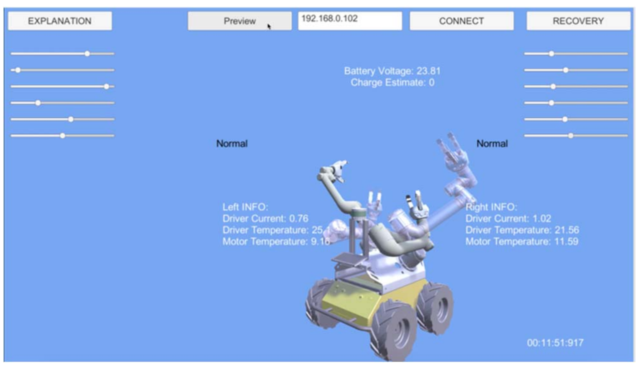

Fig. 11 A screenshot of the DT GUI shows that the "Ghost Twin" of Husky was being used for planning the upcoming joint positions of the UR5 arms

making. The "RECOVERY" button at the top right corner would immediately terminate the mission, resulting in the return of the Husky robotic system to its base station to initiate further repair actions. Once the "RECOVERY" button was pressed, the connected Raspberry Pi 3 received a command, which was translated to a ROS topic.

Therefore, users could have a complete and direct interaction with robotic systems utilizing this developed DT framework. For the control and preview function, users could make use of the "ghost twin" and sliders from the GUI to plan the next position of both arms, as shown in Fig. 11. Each slider controls a joint angle of the corresponding robotic arm, and the "ghost twin" displays the upcoming position planned by users with the sliders. Commands are sent to the edge-computing device once users make a final decision on upcoming actions, with the behavior of the physical robot driven by the updated plan from the GUI.

IV. Discussion

A DT is a new and important concept in Industry 4.0, with limitations identified where existing DT frameworks are unable to exchange run-time data between the virtual model and the physical asset, no bi-directional data stream has been built to allow full synchronization. Operators can only plan and preview the behavior of the asset on the virtual model, but it is not possible to commit commands to the physical asset. This paper has discussed three use cases that identify that a DT is suitable for both infrastructure and robotics, due to the shared requirement for health monitoring. It has been demonstrated that this DT framework provides two communication channels of data and teleoperation commands exchange between the physical and virtual environment, ensuring both sides are synchronized.

In use case 1, the embedded system illustrates that the operator can remotely monitor and control the embedded systems within infrastructure through the internet with the DT framework and GUI. The synchronization of multiple sensors and actuators allows the DT interface to display real world information, the work environment and the embedded system infrastructure as a whole, allowing for synchronization of multiple sensors in near to real-time. This enables the operator of the DT to easily and quickly identify any problems within a plant or environment, leading to decreased operational risk. In use case 2, the FMCW sensor provides inspection engineers access to previously inaccessible information on the surface and subsurface integrity of a wind turbine blade. The data collected from the FMCW radar is incorporated into the DT, providing the remote operator a holistic view of any faults on the wind turbine blade. This information would have previously been difficult to visualize for the asset operator or inspection contractor, especially as wind turbines move further offshore. Application of the 3D model in the DT enables the asset management team to visualize faults at an earlier stage, enabling an effective maintenance intervention plan to be put in place. In use case 3, the DT increases safety-compliance of robotic platforms, aiding trust between an operator and robotic application, resulting in a resilient robotic system. This DT framework improves the interaction between human and robots using the GUI application, which simplifies the ROS topic subscription and publication with an edge-computing device. The mission planning and preview can be performed with the "ghosting" feature. This feature of the DT can be used on platforms with UR5 arms, or driven by ROS, and ensures safety via a simulation of the arm executed before being committed to that action on the real-world robotic platform. The purpose of this feature is to implement a collision alert via the "ghost twin", minimizing any risk of collision in the real world. This action provides warnings to end-users when the planned position of manipulator arms contacts with other objects or asset infrastructure. The application of inverse kinematics to the GUI is a subject of active and ongoing research, which could fulfil the concept of easy control and interaction via drag and drop of the end effector.

V. Conclusion and Future Work

Data driven asset monitoring is an evolving area of research with advances in sensing technologies, data analytics and digital technologies. With increasing complexity in systems and growing dependencies across networks of systems, it is vital that tools are developed to support the aggregated integration of information and data from distributed monitoring systems. Our research shows that the developed DT framework can improve human-asset interaction, requiring less specialist knowledge from end users. Thus, DT technologies offer a direct and intuitive method to end users concerned with operational decision support. The virtual space uses animation to reflect the run-time information on the physical assets and the operating environment. It also provides intuitive interfaces with displayed colors to reflect transition into different states of health. Our interoperable DT framework will be further evaluated in other real-world scenarios, such as robotic platforms, for proving that operators can fully interact with their physical assets through the internet and the visualized virtual workspace.

ACKNOWLEDGMENT

The research reported within this paper has been supported by the Engineering Physical Sciences Research Council (EPSRC) Offshore Robotics for the Certification of Assets (ORCA). Industrial support from MicroSense Technologies Ltd (MTL) in the provision of their patent radar technology is also acknowledged as a vital contribution to this research.

REFERENCES

[1] J. Monteiro, J. Barata, M. Veloso, L. Veloso, and J. Nunes, “Towards Sustainable Digital Twins for Vertical Farming,” in *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*, 2018, pp. 234–239, doi: 10.1109/ICDIM.2018.8847169.

[2] J. Vachálek, L. Bartalský, O. Rovný, D. Šišmišová, M. Morháč, and M. Lokšík, *The Digital Twin of an Industrial Production Line Within the Industry 4.0 Concept*. .

[3] H. Zipper and C. Diedrich, “Synchronization of Industrial Plant and Digital Twin,” in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019, pp. 1678–1681, doi: 10.1109/ETFA.2019.8868994.

[4] M. Borth, J. Verriet, and G. Muller, “Digital Twin Strategies for SoS 4 Challenges and 4 Architecture Setups for Digital Twins of SoS,” in *2019 14th Annual Conference System of Systems Engineering (SoSE)*, 2019, pp. 164–169, doi: 10.1109/SYSOSE.2019.8753860.

[5] P. Wu, M. Qi, L. Gao, W. Zou, Q. Miao, and L. Liu, “Research on the Virtual Reality Synchronization of Workshop Digital Twin,” in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, 2019, pp. 875–879, doi: 10.1109/ITAIC.2019.8785552.

[6] S. Jo, D. Park, H. Park, Y. Kwak, and S. Kim, “Energy Planning of Pigsty Using Digital Twin,” in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, 2019, pp. 723–725, doi: 10.1109/ICTC46691.2019.8940032.

[7] J. Liu *et al.*, “Dynamic Evaluation Method of Machining Process Planning Based on Digital Twin,” *IEEE Access*, vol. 7, pp. 19312–19323, 2019, doi: 10.1109/ACCESS.2019.2893309.

[8] Y. Ma, H. Zhou, H. He, G. Jiao, and S. Wei, “A Digital Twin-Based Approach for Quality Control and Optimization of Complex Product Assembly,” in *2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*, 2019, pp. 762–767, doi: 10.1109/AIAM48774.2019.00157.

[9] V. V Makarov, Y. B. Frolov, I. S. Parshina, and M. V Ushakova, “The Design Concept of Digital Twin,” in *2019 Twelfth International Conference “Management of large-scale system development” (MLSD)*, 2019, pp. 1–4, doi: 10.1109/MLSD.2019.8911091.

[10] P. Zehnder and D. Riemer, “Representing Industrial Data Streams in Digital Twins using Semantic Labeling,” in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 4223–4226, doi: 10.1109/BigData.2018.8622400.

[11] O. Maryasin, “Home Automation System Ontology for Digital Building Twin,” in *2019 XXI International Conference Complex Systems: Control and Modeling Problems (CSCMP)*, 2019, pp. 70–74, doi: 10.1109/CSCMP45713.2019.8976546.

[12] J. Scheibmeir and Y. Malaiya, “An API Development Model for Digital Twins,” in *2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2019, pp. 518–519, doi: 10.1109/QRS-C.2019.00103.

[13] K. Hyeong-su, K. Jin-Woo, S. Yun, and W. Kim, “A novel wildfire digital-twin framework using interactive wildfire spread simulator,” in *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*, 2019, pp. 636–638, doi: 10.1109/ICUFN.2019.8806107.

[14] C. Xiuyu and G. Tianyi, *Research on the Predicting Model of Convenience Store Model Based on Digital Twins*. 2018.

[15] R. Revetria, F. Tonelli, L. Damiani, M. Demartini, F. Bisio, and N. Peruzzo, “A Real-Time Mechanical Structures Monitoring System Based On Digital Twin, Iot and Augmented Reality,” in *2019 Spring Simulation Conference (SpringSim)*, 2019, pp. 1–10, doi: 10.23919/SpringSim.2019.8732917.

[16] A. Bilberg and A. A. Malik, “Digital twin driven human–robot collaborative assembly,” *CIRP Ann.*, vol. 68, no. 1, pp. 499–502, 2019, doi: https://doi.org/10.1016/j.cirp.2019.04.011.

[17] F. Biesinger, D. Meike, B. Kras, and M. Weyrich, *A Case Study for a Digital Twin of Body-in-White Production Systems General Concept for Automated Updating of Planning Projects in the Digital Factory*. 2018.

[18] Y. Gao, H. Lv, Y. Hou, J. Liu, and W. Xu, “Real-time Modeling and Simulation Method of Digital Twin Production Line,” in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, 2019, pp. 1639–1642, doi: 10.1109/ITAIC.2019.8785703.

[19] M. Eckhart, A. Ekelhart, and E. Weippl, “Enhancing Cyber Situational Awareness for Cyber-Physical Systems through Digital Twins,” in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019, pp. 1222–1225, doi: 10.1109/ETFA.2019.8869197.

[20] F. Xiang, Z. Zhi, and G. Jiang, “Digital twins technolgy and its data fusion in iron and steel product life cycle,” in *2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*, 2018, pp. 1–5, doi: 10.1109/ICNSC.2018.8361293.

[21] L. Merkle, A. S. Segura, J. T. Grummel, and M. Lienkamp, “Architecture of a Digital Twin for Enabling Digital Services for Battery Systems,” in *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*, 2019, pp. 155–160, doi: 10.1109/ICPHYS.2019.8780347.

[22] A. Stetco *et al.*, “Machine learning methods for wind turbine condition monitoring: A review,” *Renew. Energy*, vol. 133, pp. 620–635, 2019, doi: https://doi.org/10.1016/j.renene.2018.10.047.

[23] V. Kongezos and E. Jellum, “Industrial asset management strategies for the oil & gas sector,” in *IET & IAM Asset Management Conference 2012*, 2012, pp. 1–6, doi: 10.1049/cp.2012.1922.

[24] Y. Bot and D. Azoulay, “Asset maintenance simulation: The case-study of an offshore wind farm,” in *2015 Annual Reliability and Maintainability Symposium (RAMS)*, 2015, pp. 1–6, doi: 10.1109/RAMS.2015.7105158.

[25] D. Flynn and D. Herd, *Prognostic Health Management techniques for intelligent condition monitoring of offshore renewable generation assets*, vol. 2013. 2013.

[26] J. Blanche, J. Buckman, H. Lewis, D. Flynn, and G. Couples, “Frequency Modulated Continuous Wave Analysis of Dynamic Load Deformation in Geomaterials,” in *Offshore Technology Conference*, 2020, doi: 10.4043/30479-ms.

[27] J. Blanche *et al.*, “Dynamic Fluid Ingress Detection in Geomaterials using K-band Frequency Modulated Continuous Wave Radar,” *IEEE Access (Early Access)*, 2020, doi: 10.1109/ACCESS.2020.3002147.

[28] J. Blanche *et al.*, “Analysis of Sandstone Pore Space Fluid Saturation and Mineralogy Variation via Application of Monostatic K-Band Frequency Modulated Continuous Wave Radar,” *IEEE Access*, vol. 6, 2018, doi: 10.1109/ACCESS.2018.2863024.